

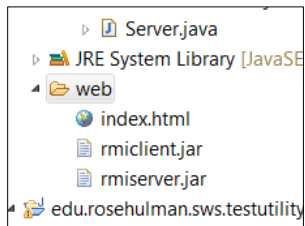
Carly Heibel and Michael Royappa

Milestone 1

CSSE477

POST Request

File Hierarchy before POST:



POST Request:

```
POST /temp.txt HTTP/1.1
content-length: 463
referer: http://localhost:8080/upload.html
accept-language: en-US,en;q=0.8
origin: http://localhost:8080
host: localhost:8080
connection: keep-alive
content-type: multipart/form-data; boundary=----WebKitFormBoundaryEzjCwl5nt4xFk9AS
cache-control: max-age=0
accept-encoding: gzip, deflate
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: HttpTestClient/1.0

"This is the content of the temp.txt file."
```

Output of POST in TestClient:

```
Connection Response
HTTP 1.1 Test Client by C.R. Rupakheti
-----
Connection Established!
Request Sent. Waiting for response ...
----- Header -----
HTTP/1.1 200 OK
date : Mon Apr 27 12:52:09 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
last-modified : Mon Apr 27 12:52:09 EDT 2015
content-length : 463
provider : Chandan R. Rupakheti
connection : Close
content-type : text/plain
----- Body -----
"This is the content of the temp.txt file."
-----
Socket Disconnected!
```

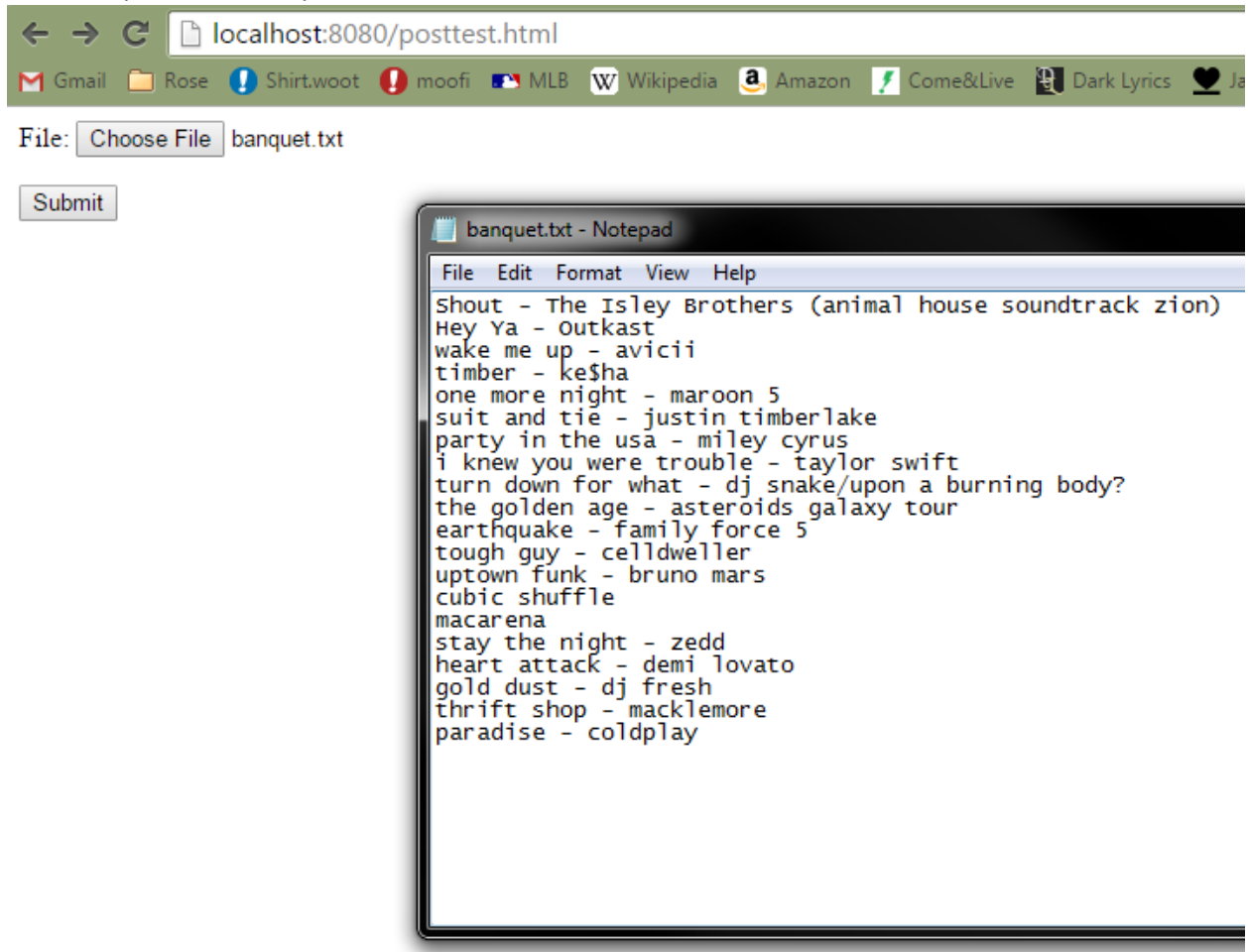
File Hierarchy after running POST Request:

```
JRE System Library [JavaS
└─ web
   ├── index.html
   ├── rmiclient.jar
   ├── rmiserver.jar
   └── temp.txt
```

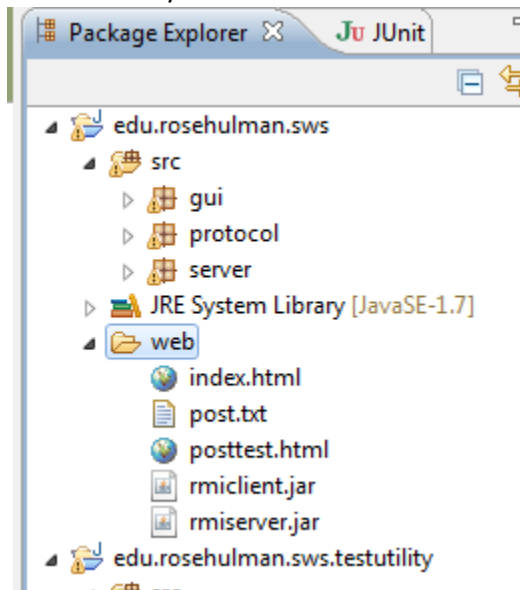
File contents after successful POST:

```
ConnectionHa...  temp.txt  Protocol.java  H
"This is the content of the temp.txt file."
```

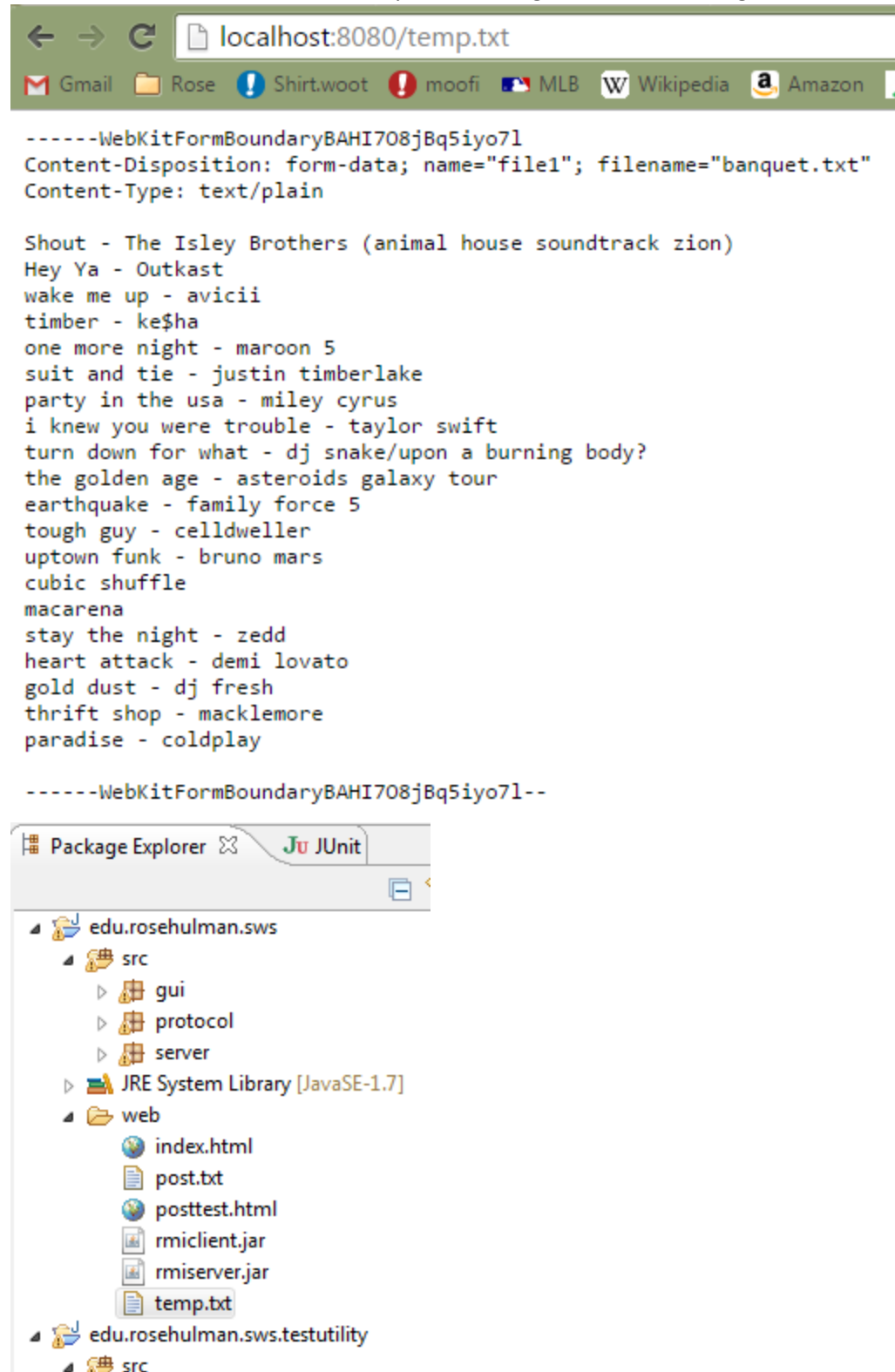
File to be posted to /temp.txt in browser:



Web directory before POST:



File contents in browser and directory after clicking "Submit" and firing POST:



The image shows two screenshots. The top screenshot is a web browser window displaying the contents of `localhost:8080/temp.txt`. The browser's address bar shows the URL, and the page content is a multipart/form-data POST response. It starts with a boundary line `-----WebKitFormBoundaryBAHI708jBq5iyo7l`, followed by headers: `Content-Disposition: form-data; name="file1"; filename="banquet.txt"` and `Content-Type: text/plain`. The main body of the response is a list of songs and artists, such as "Shout - The Isley Brothers", "Hey Ya - Outkast", "wake me up - avicii", "timber - ke\$ha", "one more night - maroon 5", "suit and tie - justin timberlake", "party in the usa - miley cyrus", "i knew you were trouble - taylor swift", "turn down for what - dj snake/upon a burning body?", "the golden age - asteroids galaxy tour", "earthquake - family force 5", "tough guy - celldweller", "uptown funk - bruno mars", "cubic shuffle", "macarena", "stay the night - zedd", "heart attack - demi lovato", "gold dust - dj fresh", "thrift shop - macklemore", and "paradise - coldplay". The response ends with another boundary line `-----WebKitFormBoundaryBAHI708jBq5iyo7l--`.

The bottom screenshot is an IDE window showing the "Package Explorer" view. It displays a project structure for `edu.rosehulman.sws`. The project has a `src` folder containing `gui`, `protocol`, and `server` subfolders. There is also a `JRE System Library [JavaSE-1.7]` and a `web` folder. The `web` folder contains several files: `index.html`, `post.txt`, `posttest.html`, `rmiclient.jar`, `rmiserver.jar`, and `temp.txt`. Below the main project, there is another project entry `edu.rosehulman.sws.testutility` with a `src` folder.

PUT Request

PUT Request:

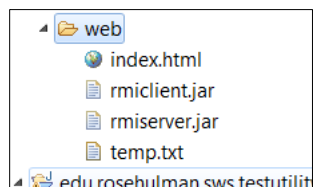
```
PUT /temp.txt HTTP/1.1
content-length: 60
referer: http://localhost:8080/upload.html
accept-language: en-US,en;q=0.8
origin: http://localhost:8080
host: localhost:8080
connection: keep-alive
content-type: multipart/form-data; boundary=----WebKitFormBoundaryEzjCwl5nt4xFk9AS
cache-control: max-age=0
accept-encoding: gzip, deflate
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: HttpTestClient/1.0
```

"PUT REQUEST: This is the content of the temp.txt file."

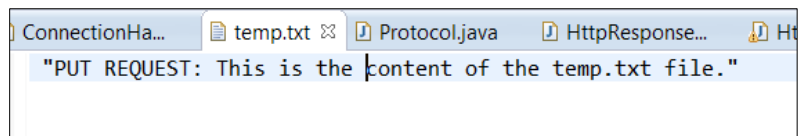
Output of PUT Request in TestClient (deleted temp.txt, so it should re-create it):

```
HTTP/1.1 200 OK
date : Mon Apr 27 13:16:31 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
last-modified : Mon Apr 27 13:16:31 EDT 2015
content-length : 60
provider : Chandan R. Rupakheti
connection : Close
content-type : text/plain
----- Body -----
"PUT REQUEST: This is the content of the temp.txt file."
Socket Disconnected!
```

File Hierarchy after running PUT with no existing temp.txt:



Contents of new temp.txt:



PUT Request after file already exists:

```

Request Sent. Waiting for response ...
----- Header -----
HTTP/1.1 200 OK
date : Mon Apr 27 13:17:48 EDT 2015
server : SimpleWebServer(SWS)/1.0.0 (Windows 7/6.1/x86)
last-modified : Mon Apr 27 13:17:48 EDT 2015
content-length : 120
provider : Chandan R. Rupakheti
connection : Close
content-type : text/plain
----- Body -----
"PUT REQUEST: This is the content of the temp.txt file."  "AAAAAAAAAAAAAAAAAAAA"
Socket Disconnected!

```

```

ConnectionHa...  temp.txt  Protocol.java  HttpResponse...  HttpClient.java  PUTRequest.ja
"PUT REQUEST: This is the content of the temp.txt file."  "AAAAAAAAAAAAAAAAAAAA"

```

File Structure doesn't change:

```

web
├── index.html
├── rmiclient.jar
├── rmiserver.jar
└── temp.txt

```

DELETE Request

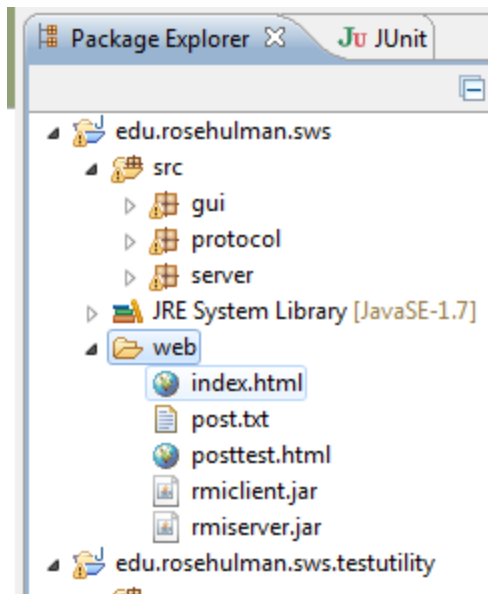
Directory before DELETE:

```

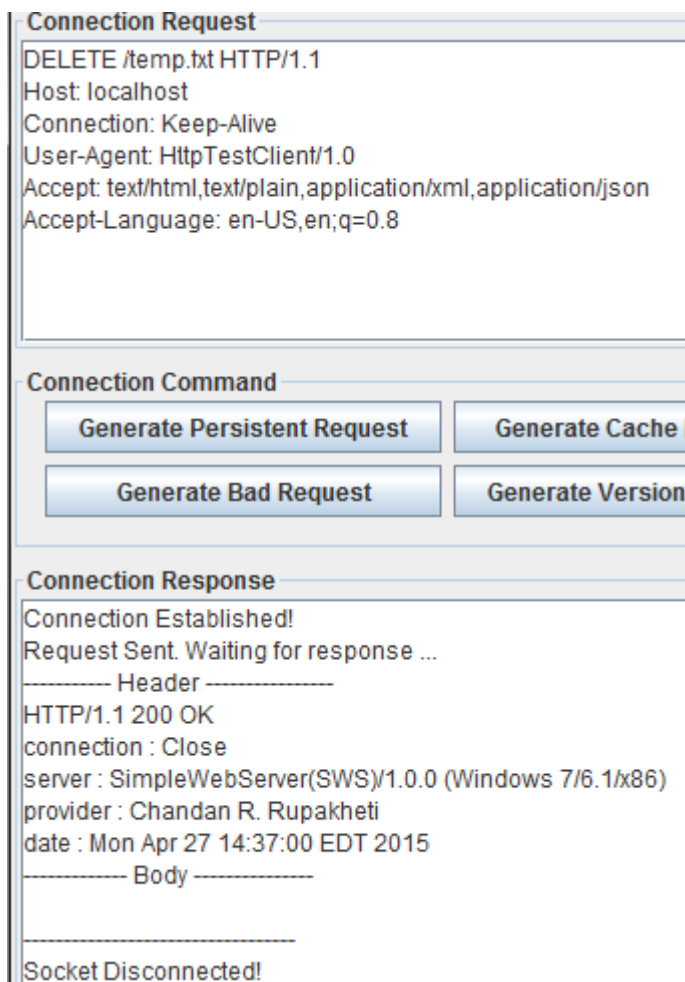
Package Explorer  JUnit
└── edu.rosehulman.sws
    ├── src
    │   ├── gui
    │   ├── protocol
    │   └── server
    ├── JRE System Library [JavaSE-1.7]
    ├── web
    │   ├── index.html
    │   ├── post.txt
    │   ├── posttest.html
    │   ├── rmiclient.jar
    │   ├── rmiserver.jar
    │   └── temp.txt
    └── edu.rosehulman.sws.testutility
        └── src

```

Directory after DELETE:



Request and response text:



Improvements

We would like to implement a `HttpRequestFactory` to streamline creation of `HttpRequest` objects. We would also like to implement some more elegant method of handling the many different response cases in `GetRequest.java`.

Class Diagram

